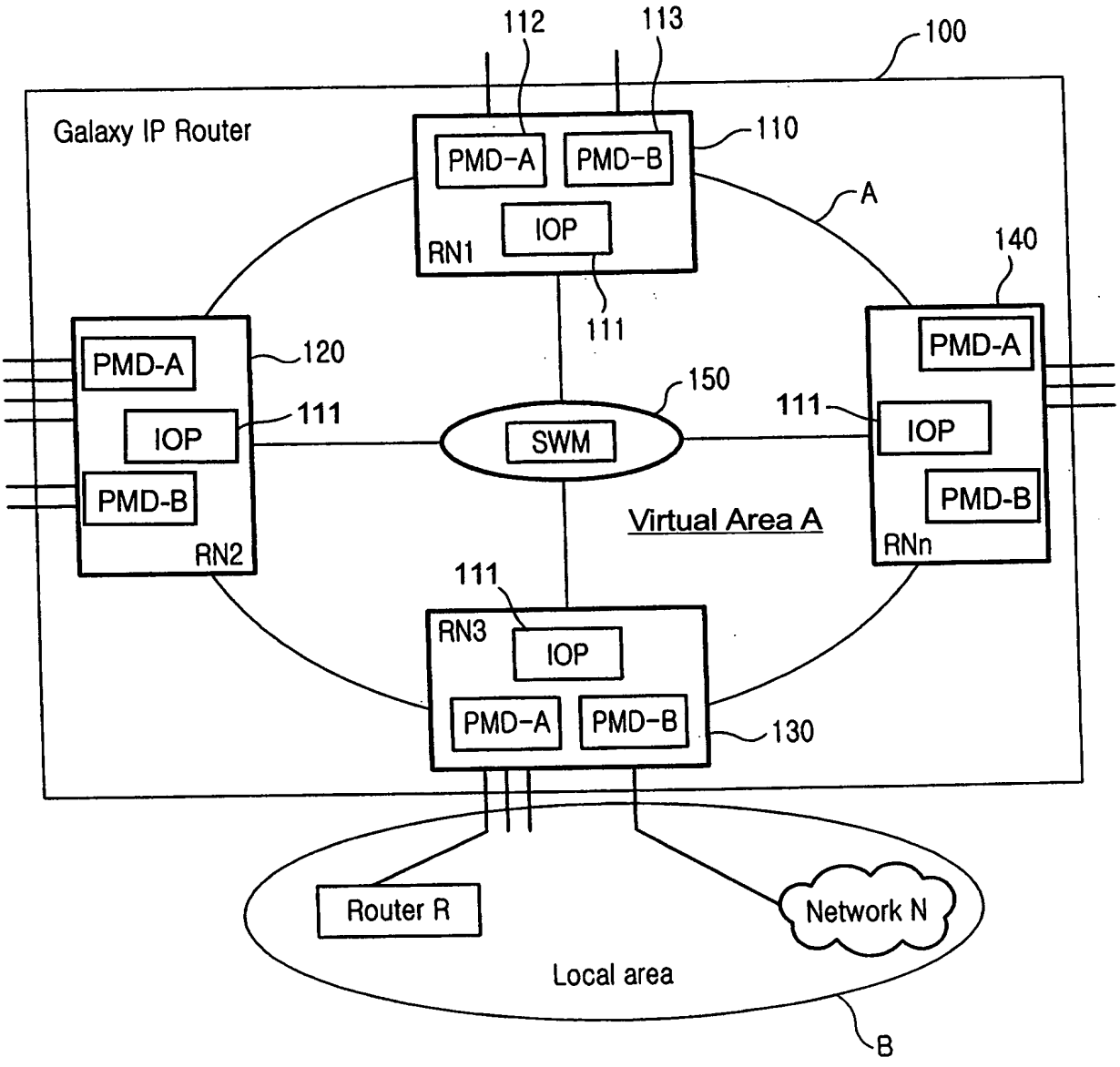

112
113
100
Galaxy IP Router
PMD-A
PMD-B
110
IOP
RN1
A
140
111
PMD-A
120
150
111
IOP
111
SWM
IOP
PMD-B
Virtual Area A
PMD-B
RN2
RNn
111
RN3
IOP
PMD-A
PMD-B
130
Router R
Network N
Local area
B

FIG. 1

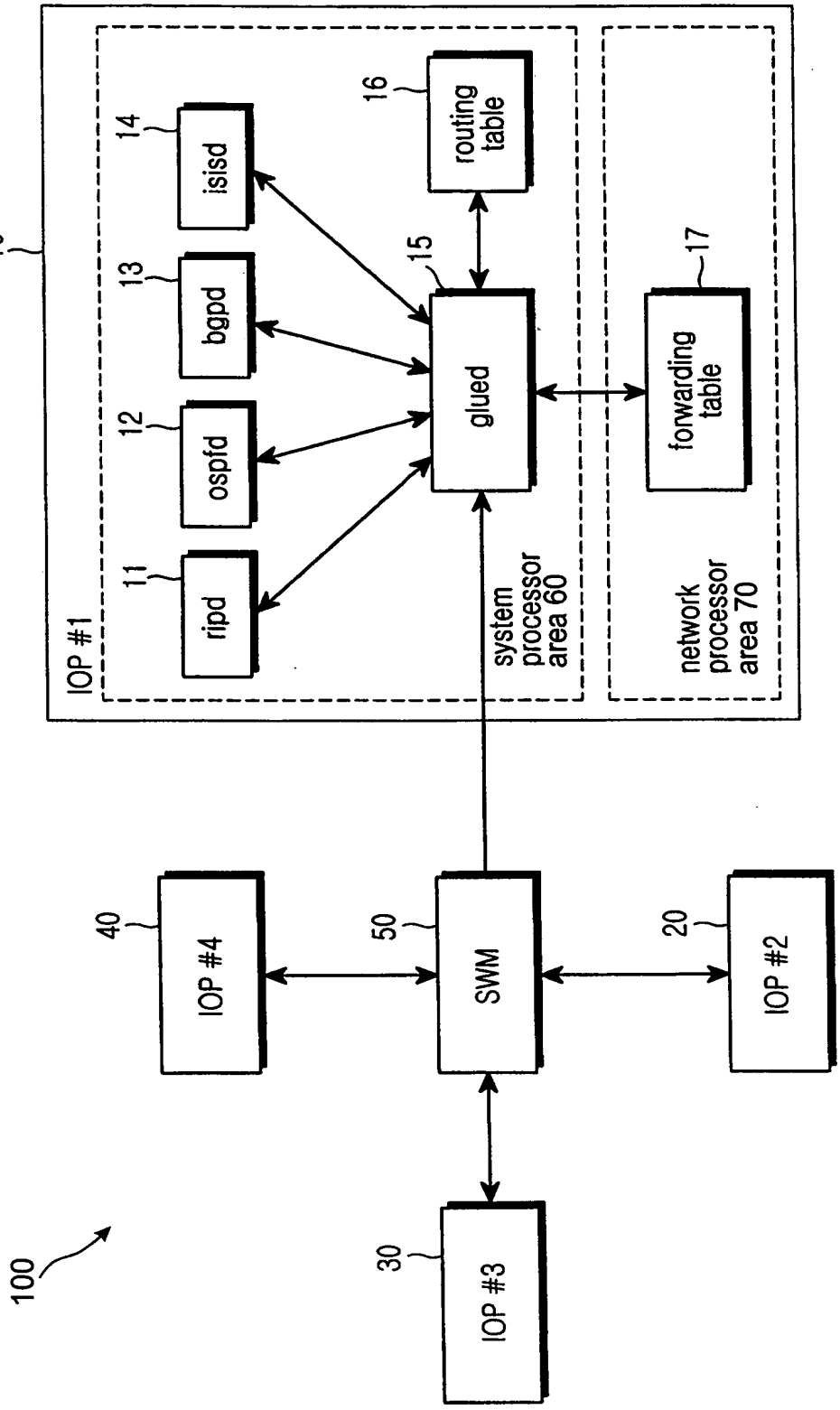

100
10
IOP #1
system processor area 60
network processor area 70
ripd
ospfd
bgpd
isisd
glued
routing table
forwarding table
11
12
13
14
15
16
17
IOP #4
IOP #3
IOP #2
SWM
40
30
20
50

FIG. 2

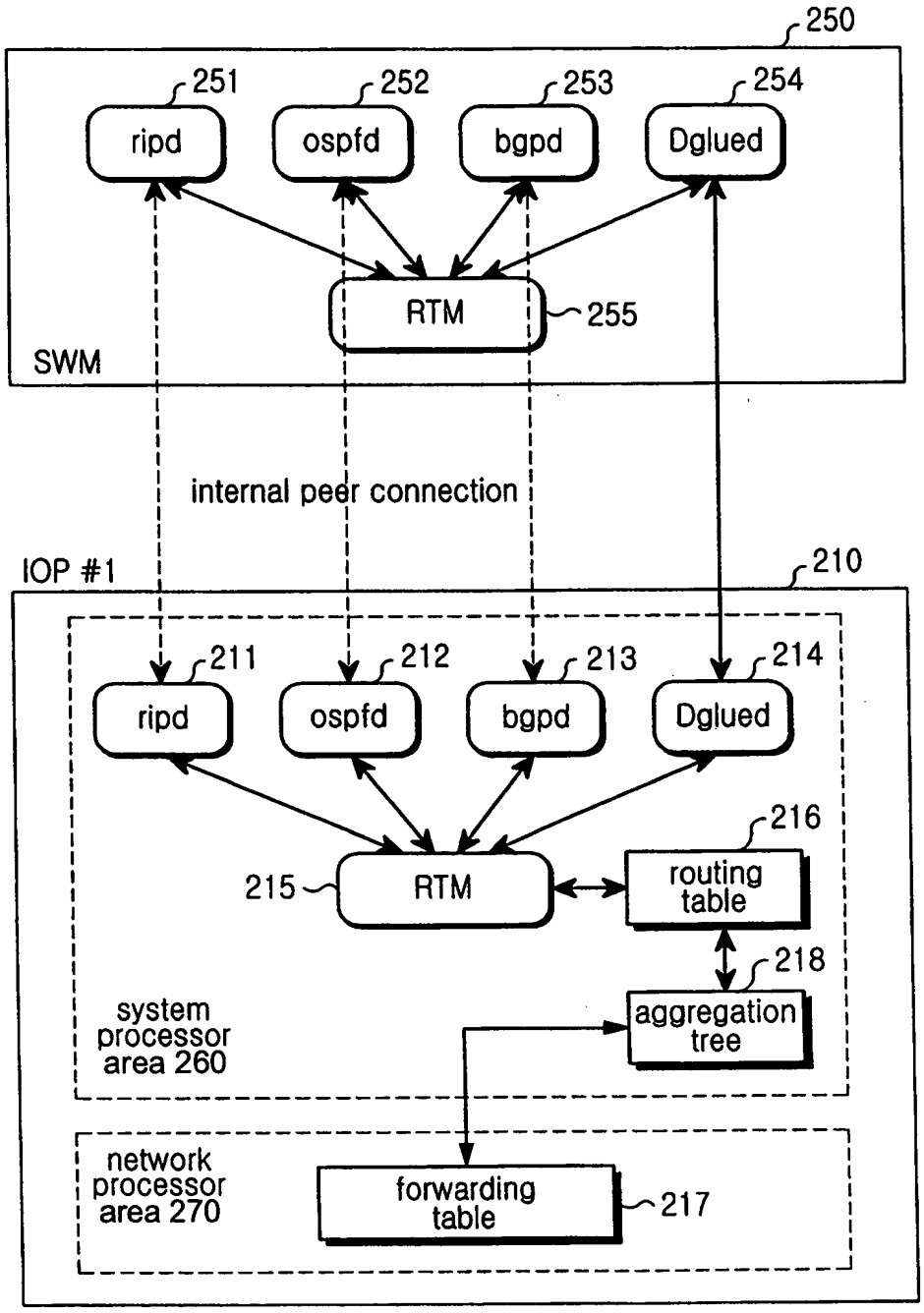

250
251
ripd
252
ospfd
253
bgpd
254
Dglued
RTM
255
SWM
internal peer connection
IOP #1
210
211
ripd
212
ospfd
213
bgpd
214
Dglued
216
215
RTM
routing table
218
aggregation tree
system processor area 260
network processor area 270
forwarding table
217

FIG. 3

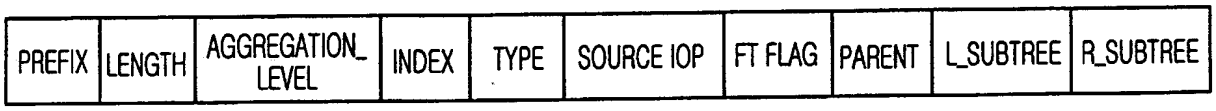

PREFIX
LENGTH
AGGREGATION_ LEVEL
INDEX
TYPE
SOURCE IOP
FT FLAG
PARENT
L_SUBTREE
R_SUBTREE

FIG. 4A

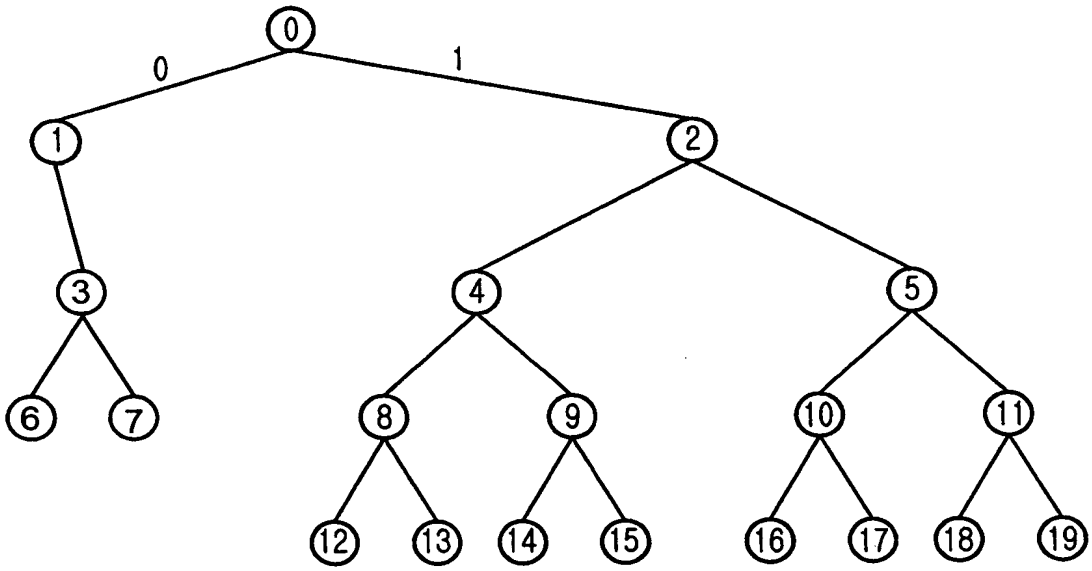


FIG. 4B

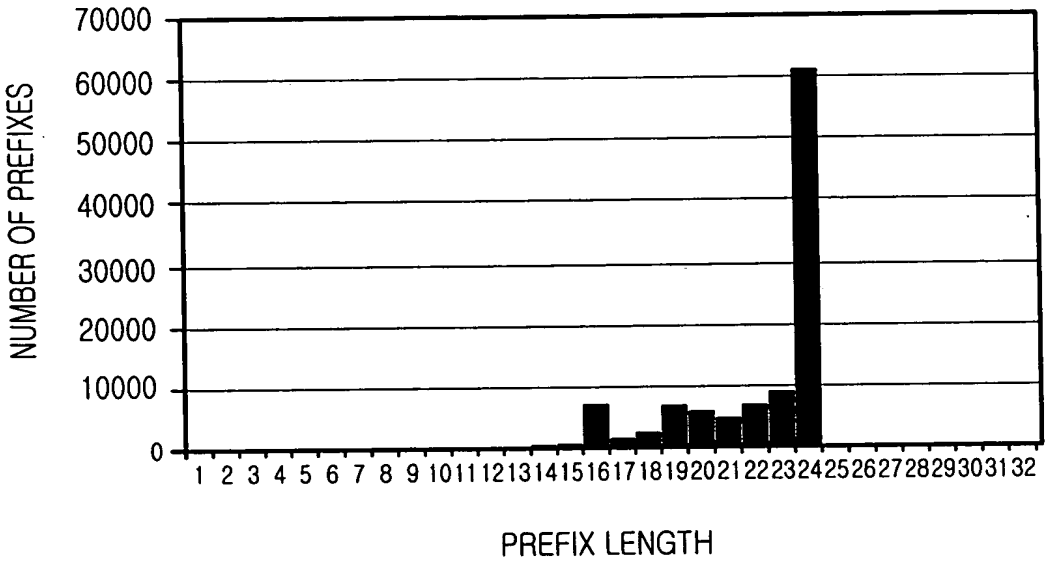

70000
60000
50000
40000
30000
20000
10000
0
NUMBER OF PREFIXES
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
PREFIX LENGTH

FIG.4C

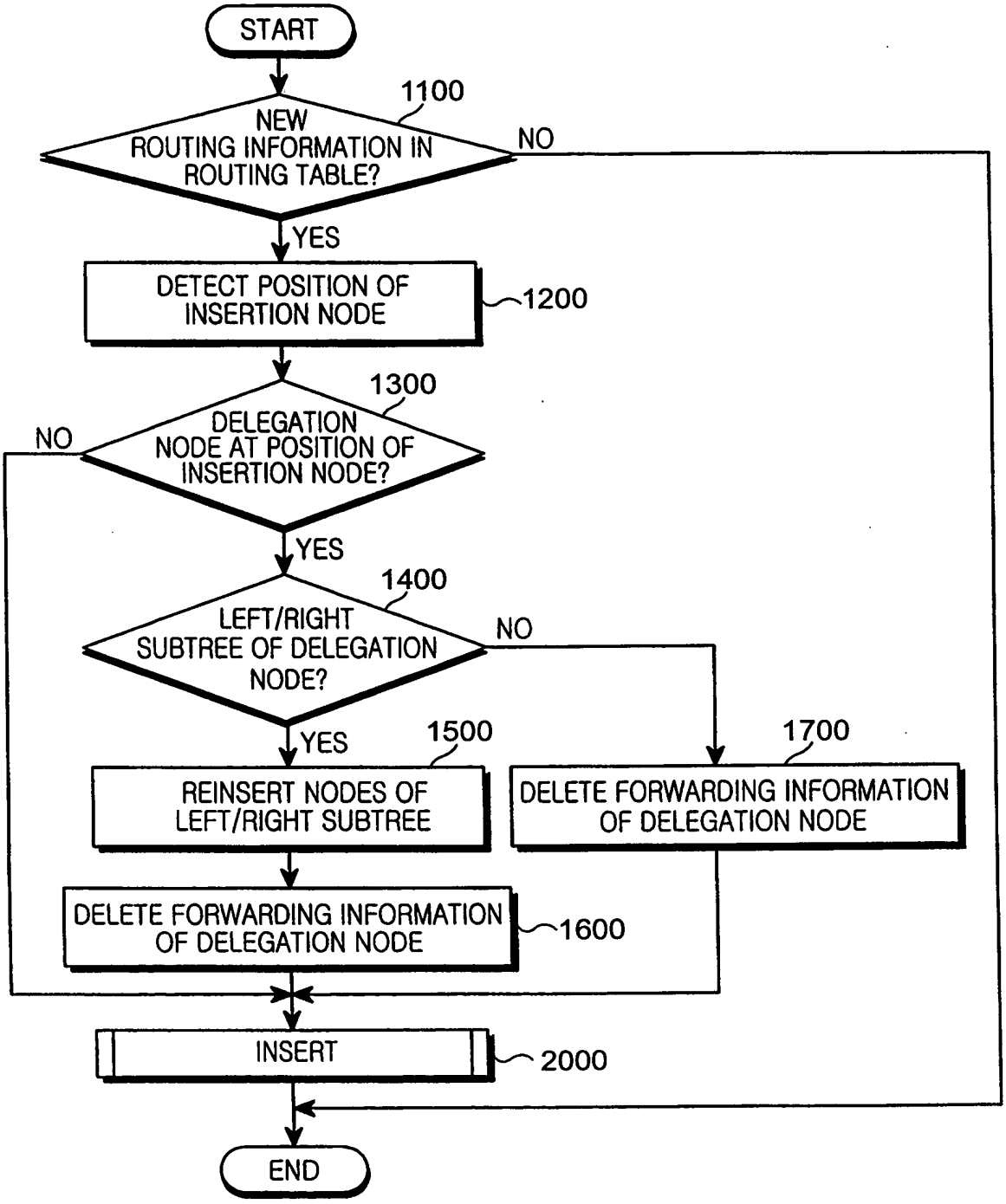

START
1100
NEW ROUTING INFORMATION IN ROUTING TABLE?
NO
YES
DETECT POSITION OF INSERTION NODE
1200
1300
DELEGATION NODE AT POSITION OF INSERTION NODE?
NO
YES
1400
LEFT/RIGHT SUBTREE OF DELEGATION NODE?
NO
YES
1500
REINSERT NODES OF LEFT/RIGHT SUBTREE
1700
DELETE FORWARDING INFORMATION OF DELEGATION NODE
DELETE FORWARDING INFORMATION OF DELEGATION NODE
1600
INSERT
2000
END

FIG. 5

FIG. 6A

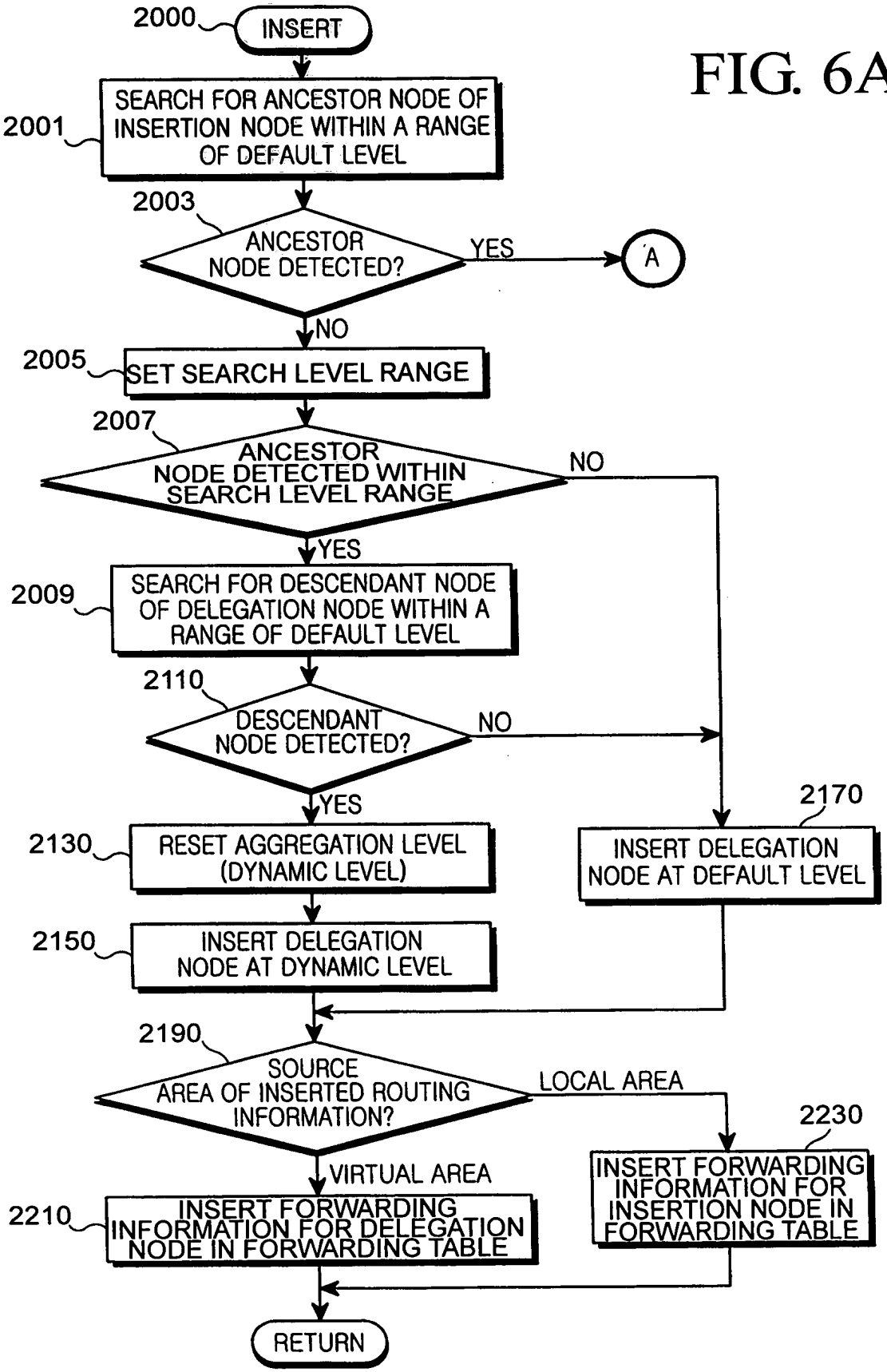
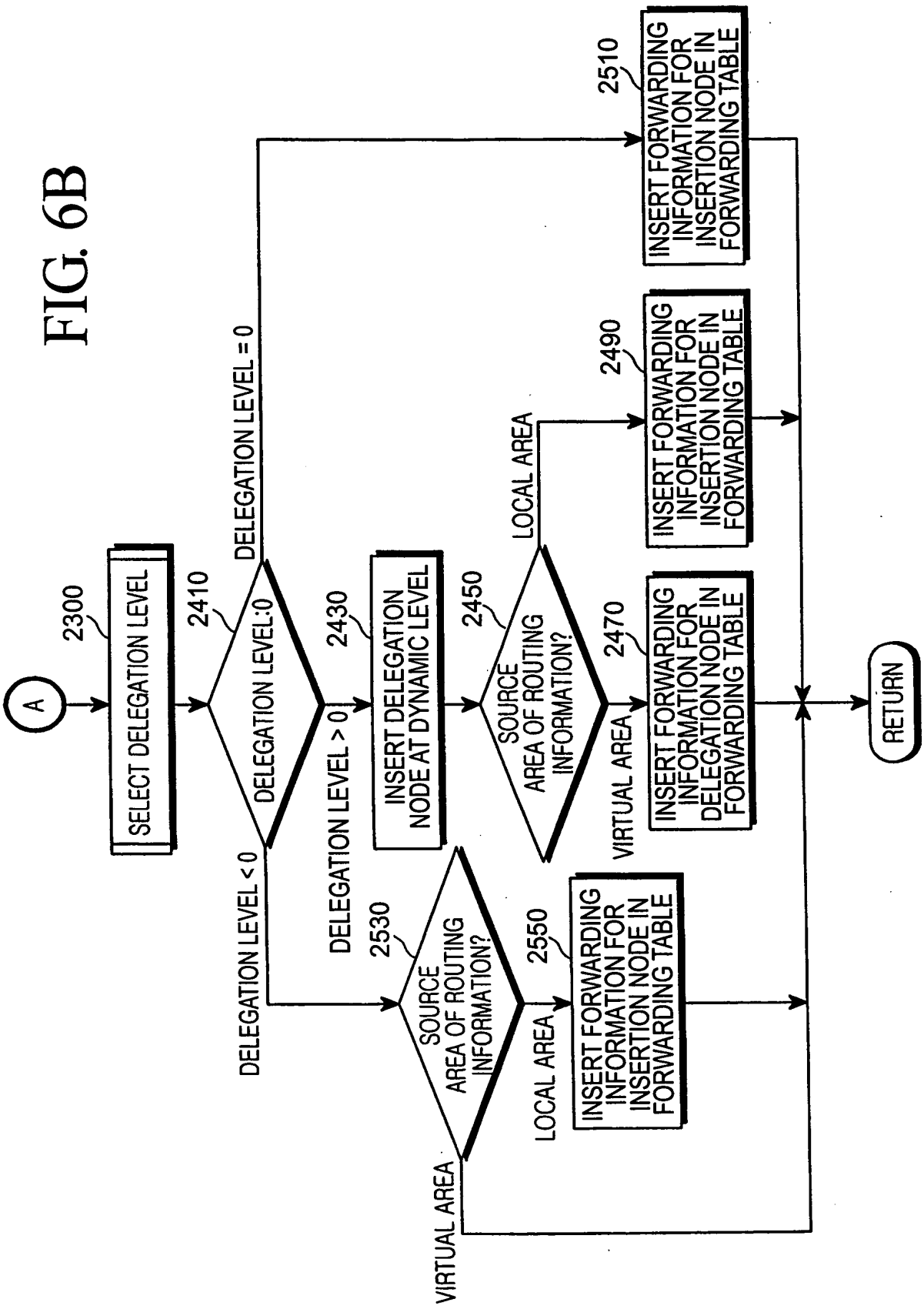

2000
INSERT
SEARCH FOR ANCESTOR NODE OF INSERTION NODE WITHIN A RANGE OF DEFAULT LEVEL
2001
2003
ANCESTOR NODE DETECTED?
YES
A
NO
2005
SET SEARCH LEVEL RANGE
2007
ANCESTOR NODE DETECTED WITHIN SEARCH LEVEL RANGE
NO
YES
2009
SEARCH FOR DESCENDANT NODE OF DELEGATION NODE WITHIN A RANGE OF DEFAULT LEVEL
2110
DESCENDANT NODE DETECTED?
NO
YES
2130
RESET AGGREGATION LEVEL (DYNAMIC LEVEL)
2150
INSERT DELEGATION NODE AT DYNAMIC LEVEL
2170
INSERT DELEGATION NODE AT DEFAULT LEVEL
2190
SOURCE AREA OF INSERTED ROUTING INFORMATION?
LOCAL AREA
VIRTUAL AREA
2210
INSERT FORWARDING INFORMATION FOR DELEGATION NODE IN FORWARDING TABLE
2230
INSERT FORWARDING INFORMATION FOR INSERTION NODE IN FORWARDING TABLE
RETURN

FIG. 6B


A
2300
SELECT DELEGATION LEVEL
2410
DELEGATION LEVEL:0
DELEGATION LEVEL = 0
2510
INSERT FORWARDING INFORMATION FOR INSERTION NODE IN FORWARDING TABLE
DELEGATION LEVEL > 0
2430
INSERT DELEGATION NODE AT DYNAMIC LEVEL
2450
SOURCE AREA OF ROUTING INFORMATION?
LOCAL AREA
2490
INSERT FORWARDING INFORMATION FOR INSERTION NODE IN FORWARDING TABLE
VIRTUAL AREA
2470
INSERT FORWARDING INFORMATION FOR DELEGATION NODE IN FORWARDING TABLE
DELEGATION LEVEL < 0
2530
SOURCE AREA OF ROUTING INFORMATION?
LOCAL AREA
2550
INSERT FORWARDING INFORMATION FOR INSERTION NODE IN FORWARDING TABLE
VIRTUAL AREA
RETURN

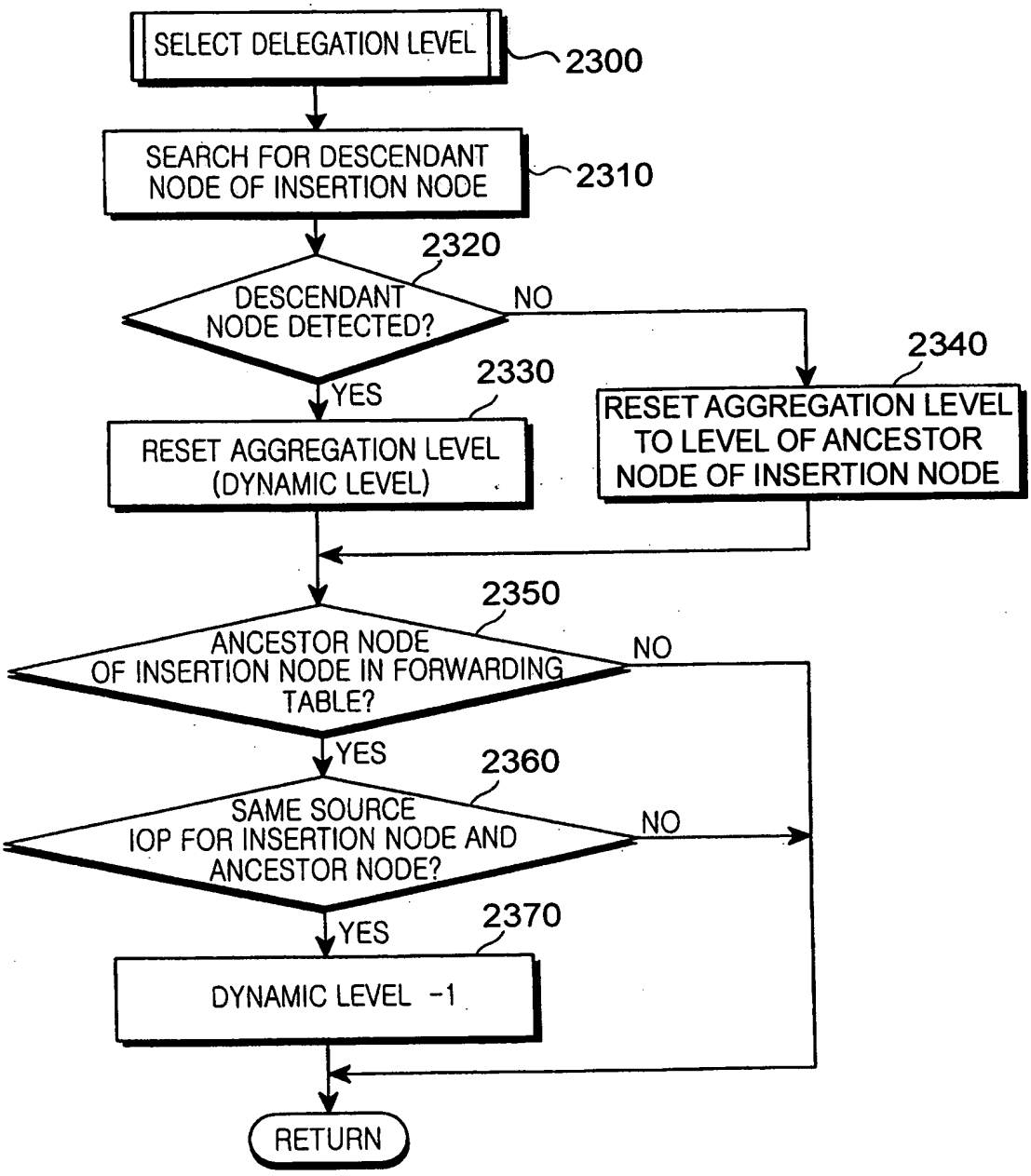
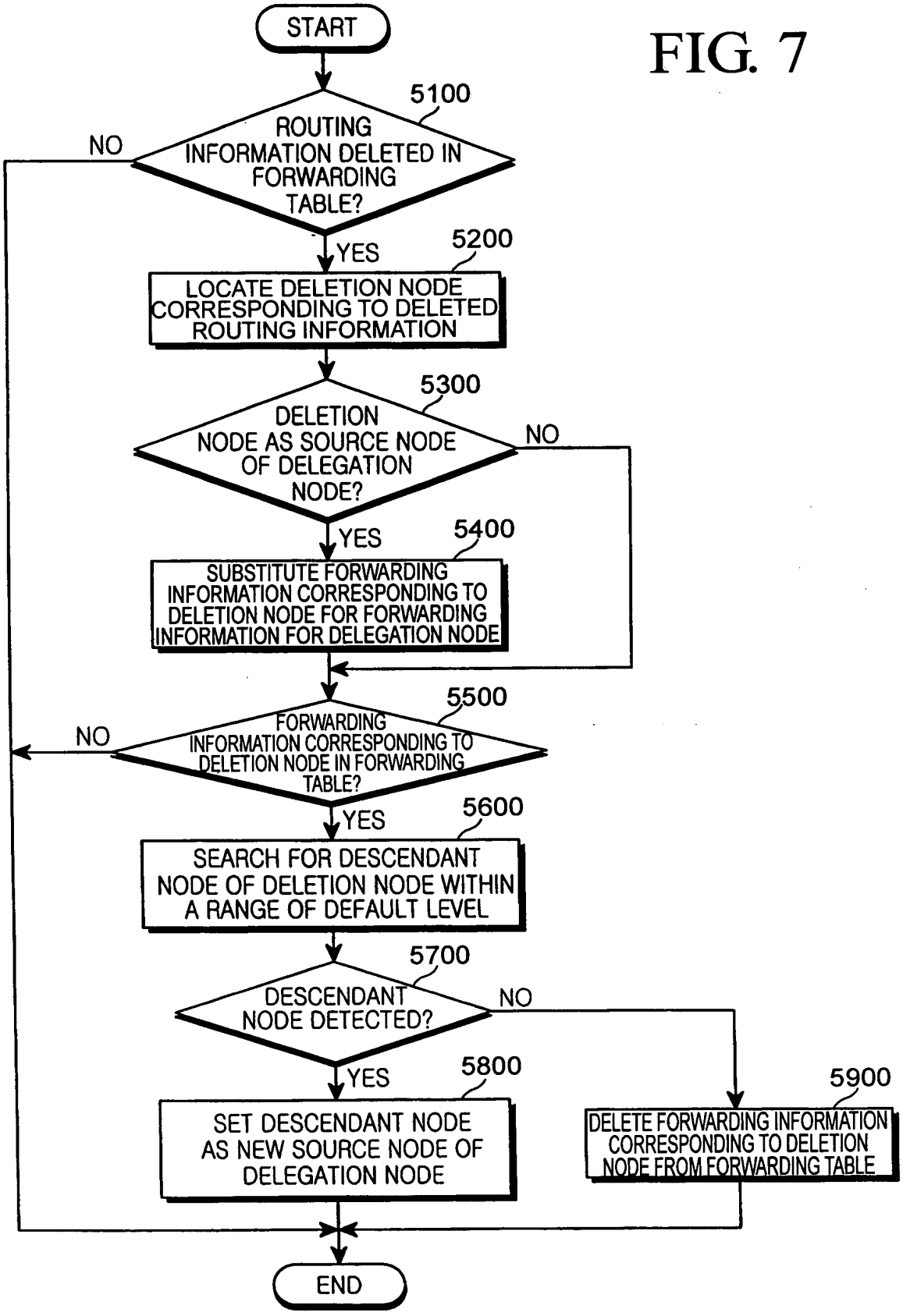

SELECT DELEGATION LEVEL
2300
SEARCH FOR DESCENDANT NODE OF INSERTION NODE
2310
2320
DESCENDANT NODE DETECTED?
NO
YES
2330
RESET AGGREGATION LEVEL (DYNAMIC LEVEL)
2340
RESET AGGREGATION LEVEL TO LEVEL OF ANCESTOR NODE OF INSERTION NODE
2350
ANCESTOR NODE OF INSERTION NODE IN FORWARDING TABLE?
NO
YES
2360
SAME SOURCE IOP FOR INSERTION NODE AND ANCESTOR NODE?
NO
YES
2370
DYNAMIC LEVEL -1
RETURN

FIG. 6C

FIG. 7


START
5100
ROUTING INFORMATION DELETED IN FORWARDING TABLE?
NO
YES
5200
LOCATE DELETION NODE CORRESPONDING TO DELETED ROUTING INFORMATION
5300
DELETION NODE AS SOURCE NODE OF DELEGATION NODE?
NO
YES
5400
SUBSTITUTE FORWARDING INFORMATION CORRESPONDING TO DELETION NODE FOR FORWARDING INFORMATION FOR DELEGATION NODE
5500
FORWARDING INFORMATION CORRESPONDING TO DELETION NODE IN FORWARDING TABLE?
NO
YES
5600
SEARCH FOR DESCENDANT NODE OF DELETION NODE WITHIN A RANGE OF DEFAULT LEVEL
5700
DESCENDANT NODE DETECTED?
NO
YES
5800
SET DESCENDANT NODE AS NEW SOURCE NODE OF DELEGATION NODE
5900
DELETE FORWARDING INFORMATION CORRESPONDING TO DELETION NODE FROM FORWARDING TABLE
END

```
Insertion (prefix) {
  local insertionnode, ancestornode, result
  /* STEP 1 */
  insertionnode := FindNode (prefix)
  /* STEP 2 */
  /* Virtual node exist in the aggregation tree */
  if (insertionnode ≠ NIL) then
    result := LeftSubTree (insertionnode)
    if (result ≠ NIL) then
     Insertion(result)
    result := RightSubTree (insertionnode)
    if (result ≠ NIL) then
      Insertion(result)
    DeleteNodeFromFT (insertionnode)
  /* STEP 3 */
  /* FindAncestorNode searches ancestor node */
 ancestornode := FindAncestorNode (insertionnode, defaultlevel)
  if (ancestornode ≠ NIL) then
    InsertWithAncestorNode (insertionnode, ancestornode)
  else
    InsertWithoutAncestorNode (insertionnode)
   }
```

FIG.8A

```
FindNode (prefix) {
  local node
  /* Search for node to be inserted */
  node := GetNode (prefix)
  /* Identity the insertion node */
  if (node ? NIL) then
        return (node)
  else
        return NIL
 }
```

FIG.8B

```
InsertWithoutAncestorNode (insertionnode) {
 local ancestornode, newdelegationnode, searchlevel,
      descendantnode, dynamiclevel
  /* Lemma 2-2 */
 searchlevel := defaultlevel*2 - 1
 ancestornode := FindAncestorNode (insertionnode, searchlevel)
  if (ancestornode = NIL) then
     dynamiclevel := defaultlevel
  else  /* FindDescendantNode finds node not written to FT  */
    descendantnode := FindDescendantNode (insertionnode, defaultlevel)
   /* Get adaptive level */
        if (descendantnode ? NIL) then
          dynamiclevel := GetNodePrefixLength (insertionnode)
              - GetDistPrefixLength (insertionnode, descendantnode)
        else
          dynamiclevel := defaultlevel
  newdelegationnode : = MakeDelegationNode (insertionnode, dynamiclevle)
  if (NodeNexthopVirtual (insertionnode) = TRUE) then
     InsertNodeToFT (newdelegationnode)
  else /* All inter-domain node must be inserted into FT */
     InsertNodeToFT (insertionnode)
}
```

FIG.8C

```
MakeDelegationNode (node, level) {
 local delegationnode
 /* Make a virtual delegation node and set node type */
   delegationnode := AllocateDelegationNode (node, level)
   SetNodeType (delegationnode) := DELEGATION
   SetDelegationNodeIndex (delegationnode, node)
   return (delegationnode)
}
```

FIG.8D

```
InsertWithAncestorNode (insertionnode, ancestornode) {
  local newdelegationnode, result, dynamiclevel
 dynamiclevel := SelectDelegationLevel (insertionnode, ancestornode)
  if (dynamiclevel > 0) then
      newdelegationnode := MakeDelegationNode (insertionnode, dynamiclevel);
   /* Resolve destination area */
      if (NodeNexthopVirtual (insertionnode) = TRUE) then
        InsertNodeToFT  (newdelegationnode)
      else
        InsertNodeToFT  (insertionnode)
  elseif (dynamiclevel = 0) then
    InsertNodeToFT  (insertionnode)
  else /* Check aggregatability of route */
    if (NodeNexthopVirtual  (insertionnode) = FALSE) then
            InsertNodeToFT  (insertionnode)
}
```

FIG.8E

```
SelectDelegationLevel (insertionnode, ancestornode) {
  local dynamiclevel, result, searchnode, descendantnode
   descendantnode := FindDescendantNode (insertionnode, GetNodeLevel (ancestornode))
   /* Get adaptive level */
   if (descendantnode ≠  NIL) then
            dynamiclevel := GetNodePrefixLength (insertionnode)
                          - GetDistPrefixLength (insertionnode, descendantnode)
   else
     dynamiclevel := GetNodesLevel (insertionnode, ancestornode)
   /* Aggregatable case */
   if (CheckNodeUpdateToFT(ancestornode) = TRUE
     && NodeSource(insertionnode) = NodeSource(ancestornode)) then
         dynamiclevel := -1

   return dynamiclevel
}
```

FIG.8F

```
Deletion (prefix) {
 local deletionnode, delegationnode,
        result, descendantnode
  /* STEP1 */
  deletionnode := FindNode (prefix)
  /* STEP2 */
 if (CheckNodeAggregate (deletionnode) = TRUE)
    then
    deletionnode := GetDelegationNode (deletionnode)
 if (CheckNodeUpdateToFT (deletionnode) = TRUE)
     then
    descendantnode := FindDescendantNode
                          (deletionnode, defaultlevel)
    /* Suppress deletion from FT */
    if (descendantnode ?  NIL) then
    SetDelegationNode (deletionnode,
        descendantnode)
     else
       DeleteNodeFromFT (deletionnode)
   }
```

FIG.8G

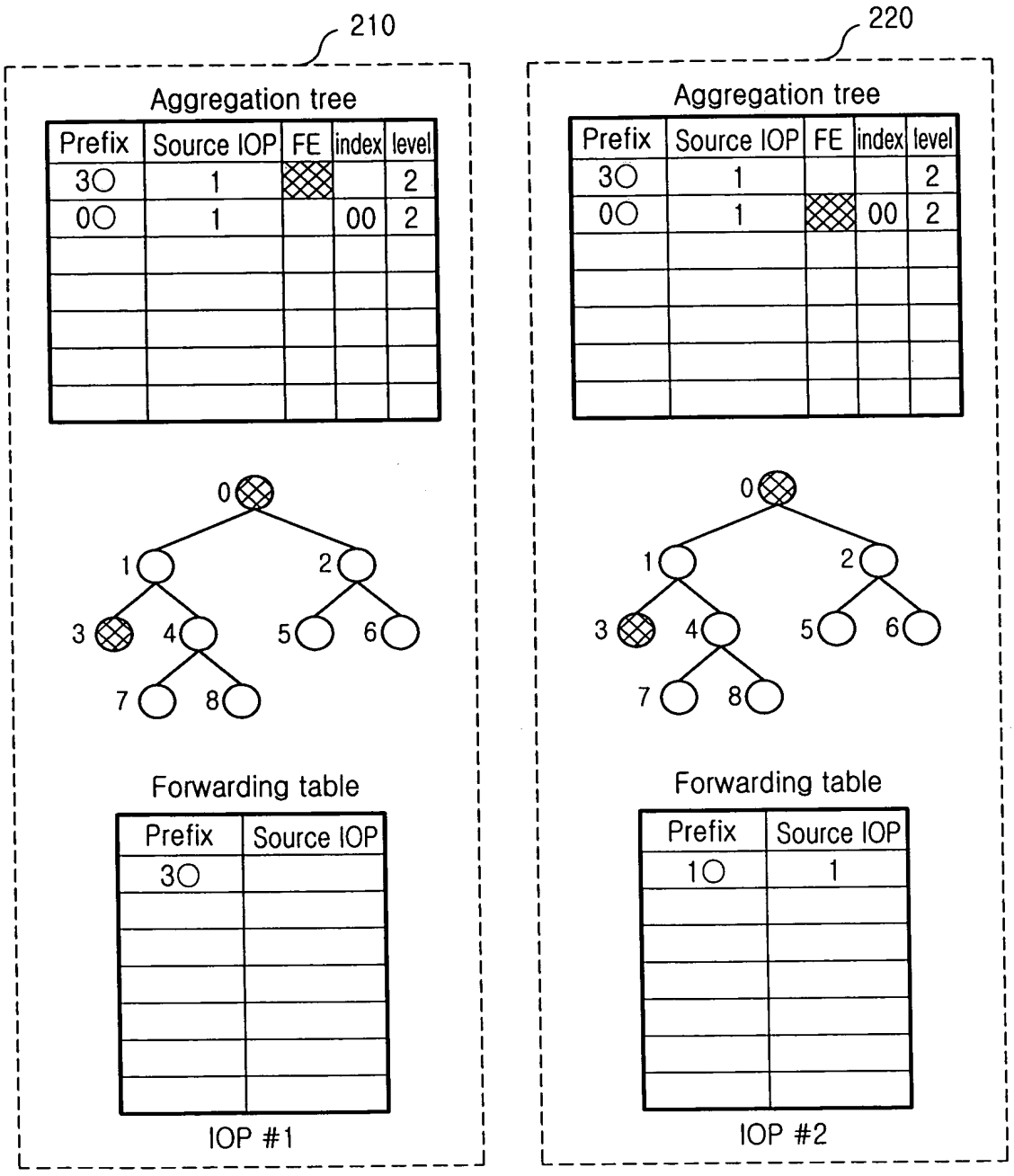

210
Aggregation tree
Prefix
Source IOP
FE
index
level
3○
1
2
0○
1
00
2
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
3○
IOP #1
220
Aggregation tree
Prefix
Source IOP
FE
index
level
3○
1
2
0○
1
00
2
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
1○
1
IOP #2

FIG.9A

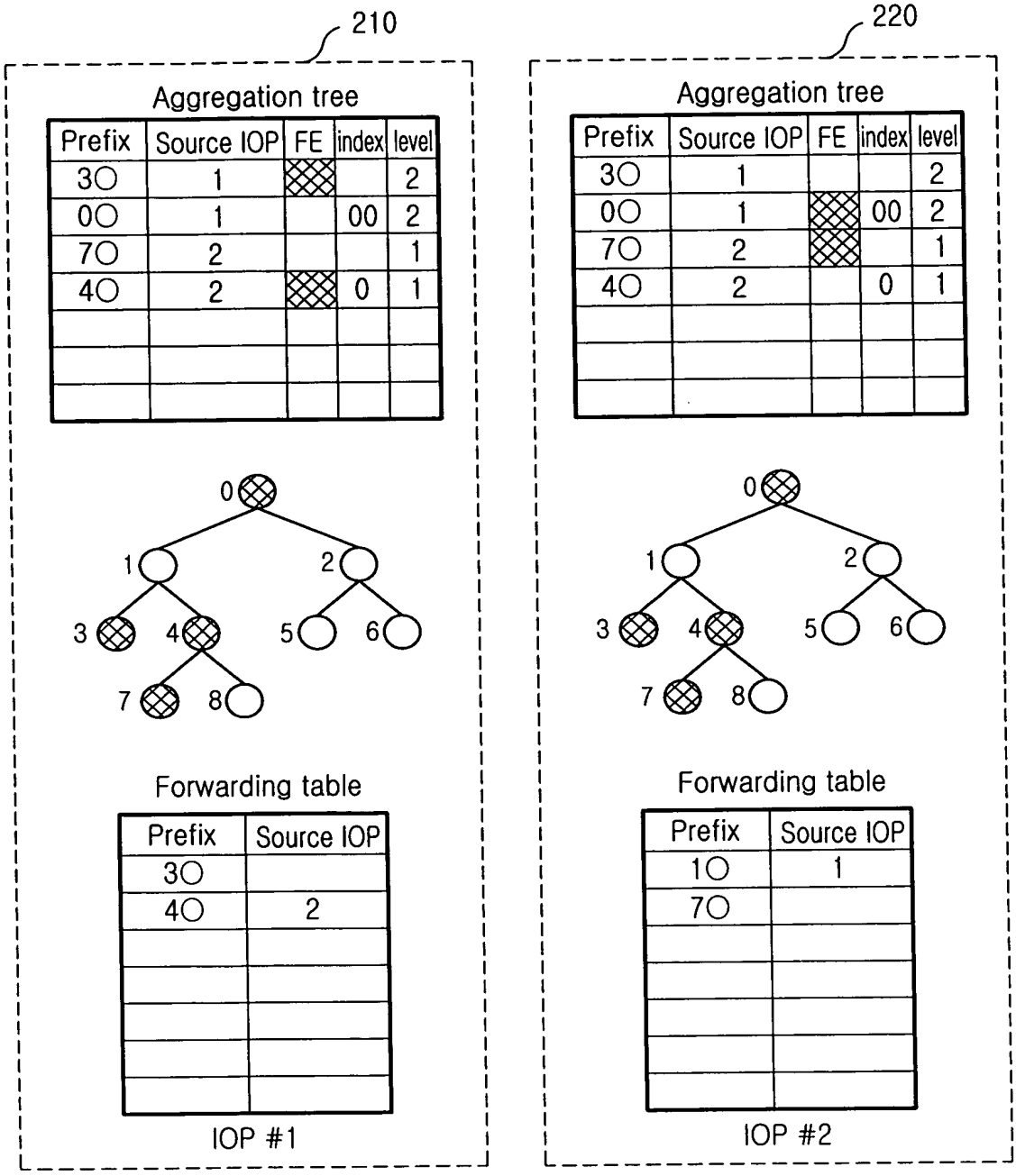

210
Aggregation tree
Prefix
Source IOP
FE
index
level
3○
1
2
0○
1
00
2
7○
2
1
4○
2
0
1
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
3○
4○
2
IOP #1
220
Aggregation tree
Prefix
Source IOP
FE
index
level
3○
1
2
0○
1
00
2
7○
2
1
4○
2
0
1
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
1○
1
7○
IOP #2

FIG.9B

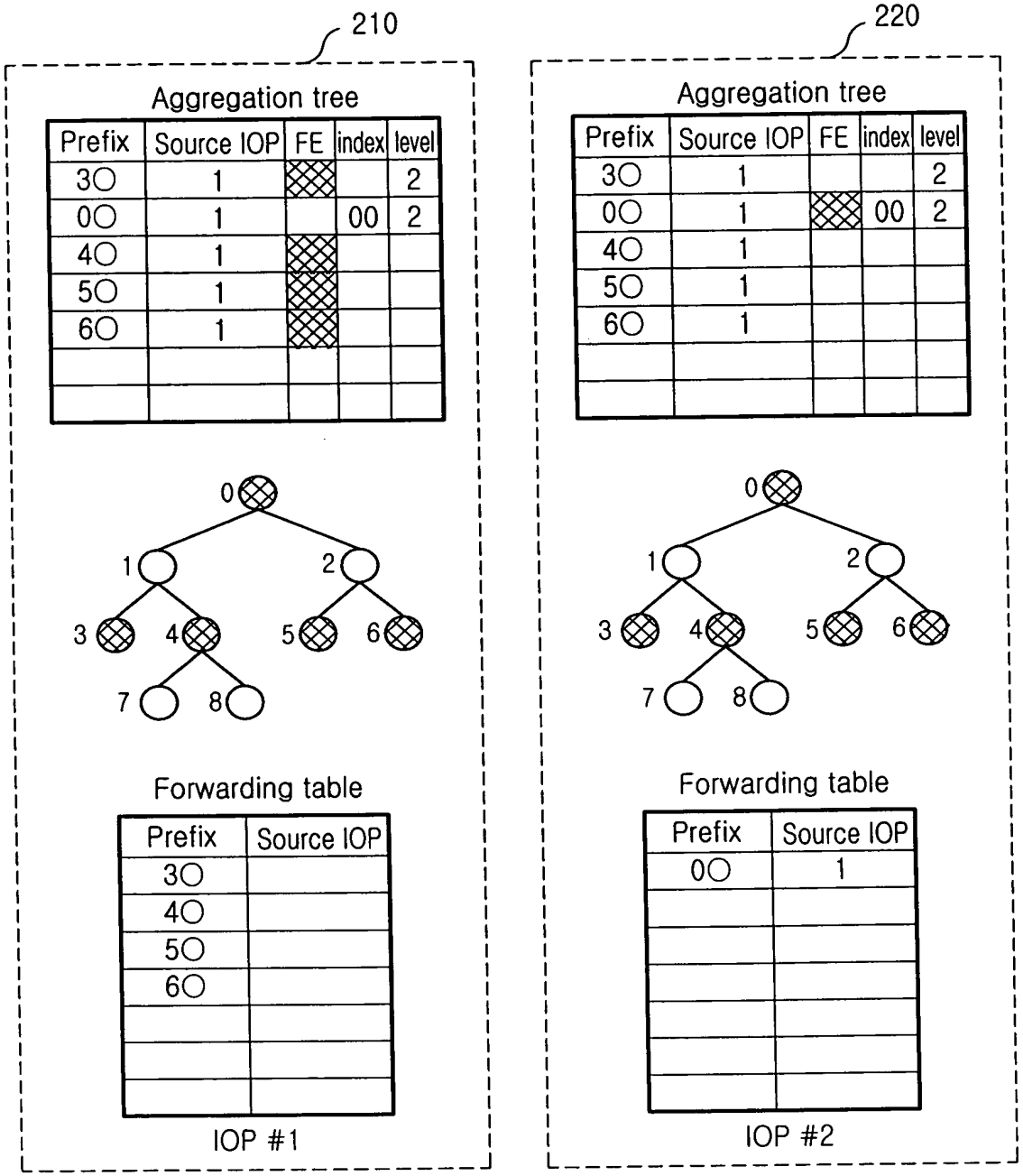

210
Aggregation tree
Prefix
Source IOP
FE
index
level
3O
1
2
0O
1
00
2
4O
1
5O
1
6O
1
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
3O
4O
5O
6O
IOP #1
220
Aggregation tree
Prefix
Source IOP
FE
index
level
3O
1
2
0O
1
00
2
4O
1
5O
1
6O
1
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
0O
1
IOP #2

FIG.10A

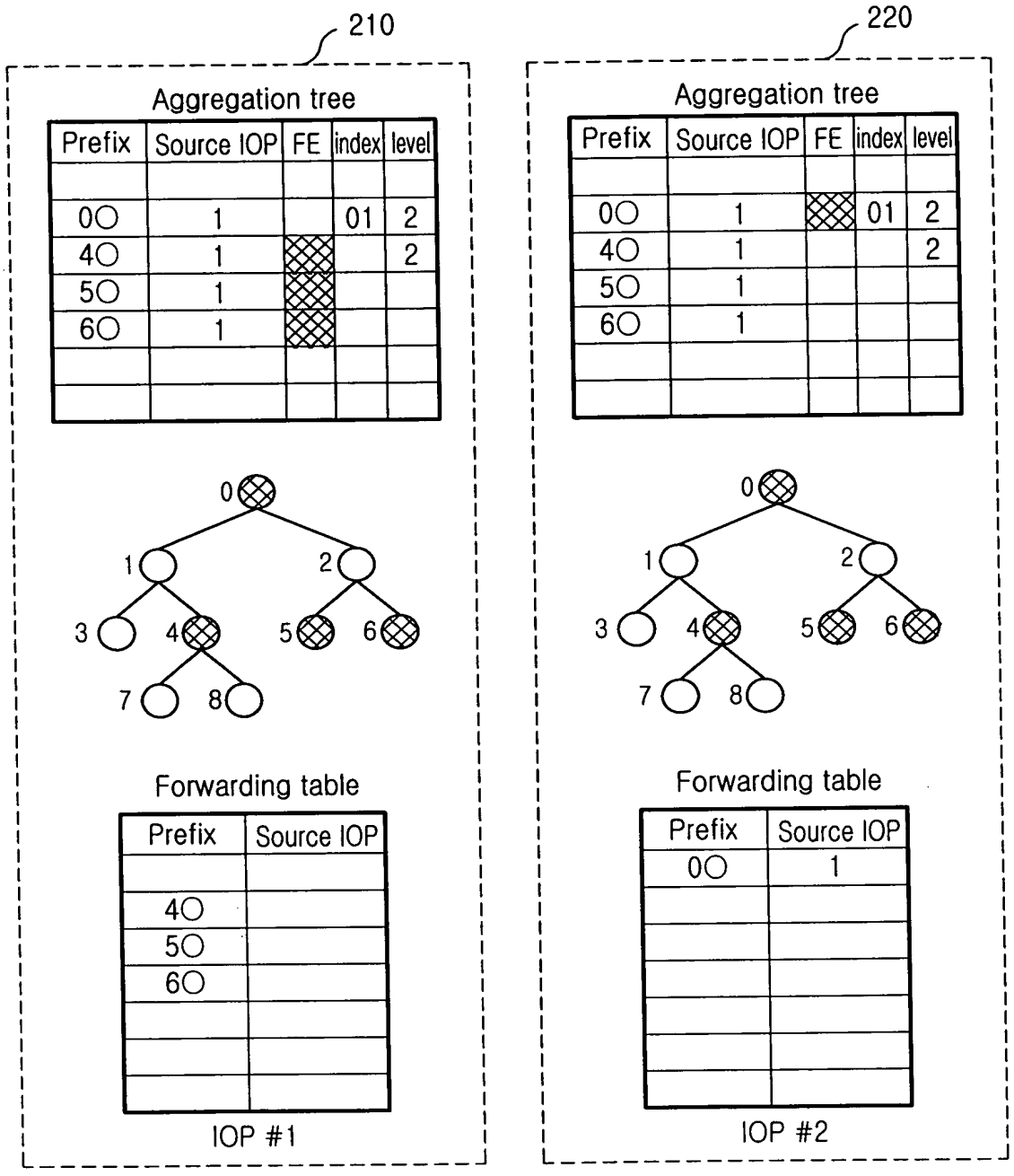

210
Aggregation tree
Prefix
Source IOP
FE
index
level
0○
1
01
2
4○
1
2
5○
1
6○
1
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
4○
5○
6○
IOP #1
220
Aggregation tree
Prefix
Source IOP
FE
index
level
0○
1
01
2
4○
1
2
5○
1
6○
1
0
1
2
3
4
5
6
7
8
Forwarding table
Prefix
Source IOP
0○
1
IOP #2

FIG.10B

Default level	Number of the entries	Number of the difference	Reduction rate
1	39,802	58	26%
2	30,574	104	43%
3	25,231	87	53%
4	20,491	118	62%
5	16,478	82	69%
6	12,554	61	77%
Original	53,632	0	0%

FIG.11A

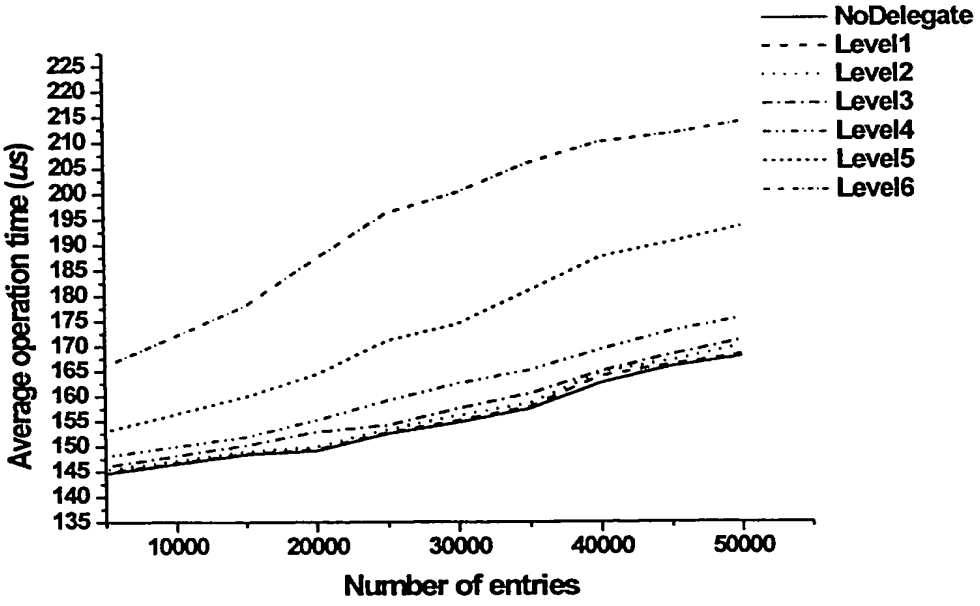


FIG.11B

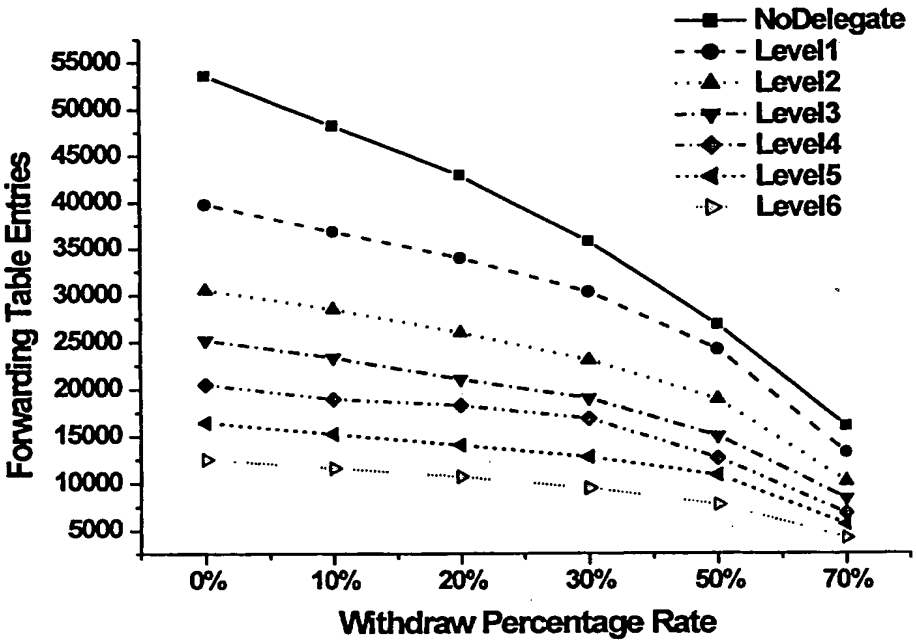


FIG.11C

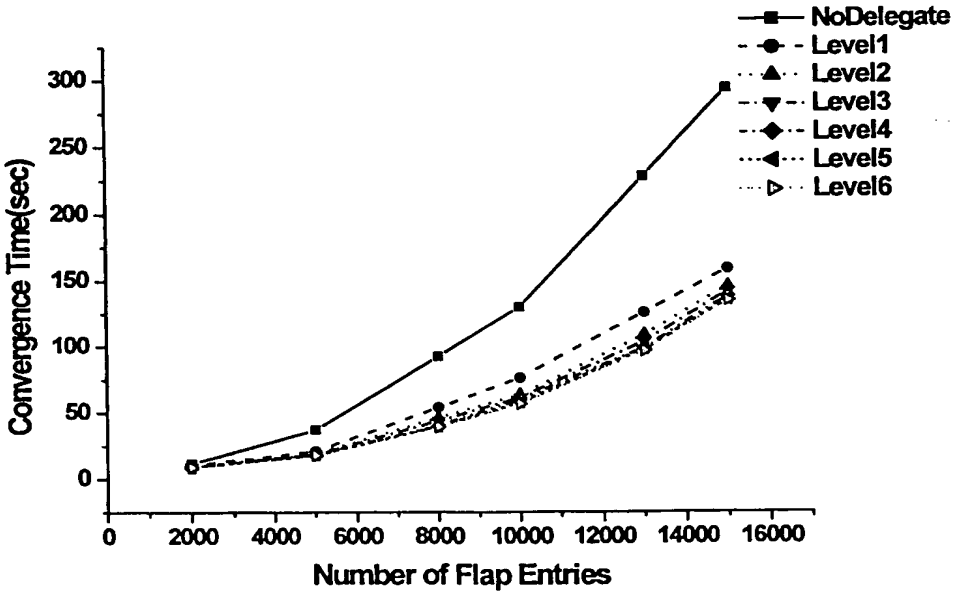


FIG.11D